Ontology Pattern Modeling for Cross-Repository Data Integration in the Ocean Sciences: The Oceanographic Cruise Example

Adila A. KRISNADHI[a,b,]*, Robert ARKO[c], Suzanne CARBOTTE[c], Cynthia CHANDLER[d], Michelle CHEATHAM[a], Timothy FININ[e], Pascal HITZLER[a], Krzysztof JANOWICZ[f], Thomas NAROCK[g], Lisa RAYMOND[d], Adam SHEPHERD[d], Peter WIEBE[d]

[a] *Wright State University, Dayton, OH, USA*
[b] *Universitas Indonesia, Depok, Jawa Barat, Indonesia*
[c] *Lamont-Doherty Earth Observatory, Columbia University, Palisades, NY, USA*
[d] *Woods Hole Oceanographic Institution, MA, USA*
[e] *University of Maryland, Baltimore County, MD, USA*
[f] *University of California, Santa Barbara, CA, USA*
[g] *Marymount University, Arlington, VA, USA*

Abstract. EarthCube is a major effort of the National Science Foundation to establish a next-generation knowledge architecture for the broader geosciences. Data storage, retrieval, access, and reuse are central parts of this new effort. Currently, EarthCube is organized around several building blocks and research coordination networks. OceanLink is a semantics-enabled building block that aims at improving data retrieval and reuse via ontologies, Semantic Web technologies, and Linked Data for the ocean sciences. Cruises, in the sense of research expeditions, are central events for ocean scientists. Consequently, information about these cruises and the involved vessels is of primary interest for oceanographers, and thus, needs to be shared and made retrievable. In this paper, we report the use of a design pattern-centric strategy to model Cruise for OceanLink data integration. We provide a formal axiomatization of the introduced pattern using the Web Ontology Language, explain design choices and discuss the planned deployment and application scenarios of our model.

Keywords. Oceanography, data integration, ontology design pattern, ontology reuse, alignment, oceanographic cruise, trajectory ontology, axiomatization, OWL

1. Introduction

Years of research in the ocean sciences, and the geosciences in general, have yielded an amount of data that is not only huge in volume, but also highly heterogeneous both in

* Corresponding Author: Adila A. Krisnadhi, Department of Computer Science and Engineering, Wright State University, 3640 Colonel Glenn Highway, Dayton, Ohio, USA; E-mail: krisnadhi.2@wright.edu.

types and formats, and scattered across distributed data repositories [1]. For individual researchers, this situation presents a difficult challenge in discovering, accessing, and integrating data for conducting scientific inquiries. Furthermore, this also introduces difficult knowledge management issues that must be overcome by the whole research community [2].

Sponsored by the National Science Foundation (NSF), the EarthCube initiative[1] brings together the US geoscience research community through a number of funded building blocks, research coordination networks, and special interest groups to establish a knowledge infrastructure crucial for enabling cross-discipline scientific endeavors. Intuitively, such an infrastructure can facilitate data discovery and integration through centralized facilities. On the other hand, it is often the case that data quality can be better ensured when local data sources and partners are made an active part of the framework. The challenge is then *how to realize such a centralized discovery framework while maintaining a decentralized nature*.

The OceanLink project[2] is an ongoing EarthCube building block aimed at tackling the aforementioned challenge specifically in ocean sciences [3]. Oceanographic research data in the US are maintained by numerous distributed online repositories, for example, the Biological and Chemical Oceanographic Data Management Office (BCO-DMO),[3] Rolling Deck to Repository (R2R) program,[4] Integrated Earth Data Applications (IEDA)[5], and the Index to Marine and Lacustrine Geological Samples (IMLGS),[6] to name a few. The lack of integrated knowledge infrastructure hampers researchers' ability to realize discovery scenarios possible only when multiple repositories are involved. For example, one may be interested in determining if the Global Multi-Resolution Topography (GMRT)[7] synthesis grid [4] contains high-resolution data from a ship's multibeam sonar in the proximity of a specified physiographic feature such as the Lomonosov Ridge, and returning the list of ship expeditions that contributed high-resolution data to those grid cells. One may then wish to determine which principal investigators and research programs are linked to those expeditions; which journal publications, meeting and/or funding awards contain thematic keywords pertaining to the physiographic feature; and which data sets and research products are available for those expeditions at each online repository. The OceanLink project has set out to facilitate such a discovery scenario, which is a vision many oceanographers would hope to see realized.

However, building an integrated knowledge discovery framework on top of those data repositories is a hugely challenging task, both socially and technically, because the data not only often do not directly align, but more than that, there are fundamental differences in modeling, leading to insufficient overlap for conducting a meaningful integration. OceanLink addresses this challenge using advances in Semantic Web technologies, particularly Linked Data [5] and Ontology Design Patterns (ODPs) [6]. The former allows the repositories to describe and expose their data in a standard syntax that is natural for linking with other data, possibly in different repositories. The latter enables a horizontal integration where semantic alignment occurs for specific

[1] http://www.earthcube.org
[2] http://www.oceanlink.org
[3] http://www.bco-dmo.org
[4] http://www.rvdata.us
[5] http://www.iedadata.org
[6] http://www.seabedsamples.org
[7] http://www.marine-geo.org/portals/gmrt/

purposes between repositories with potentially independent semantic models. Such a horizontal integration is possible through an approach based on ODPs because it is not advocating an overarching, upper-level ontology that captures a global agreement on all concepts and relationships across all data repositories, something that is often infeasible even within a single scientific domain [7]. Rather, the ODP approach is to specify a set of ontology design patterns – more precisely content patterns, each of which is simply a partial ontology that formalizes only one key notion, and to do it in such a robust way that it can be aligned with the differing representation choices that had already been made in different repositories.

One such key notion occurring across many ocean science repositories is the notion of *cruise*. Roughly, a cruise in ocean sciences, or an *oceanographic cruise*, is an expedition conducted on a vessel to the ocean or other navigable water body for particular purposes typically related to oceanographic research activities. Cruises hold a critical role in ocean sciences, because most oceanographic research activities such as field observations, data acquisition, and scientific experiments can be accomplished only when researchers gain direct access to the oceans using vessels [8]. Note that a cruise should be distinguished from the corresponding vessel as the latter is an actual physical object, whereas the former concerns not just the vessel, but also the corresponding activities carried out while the vessel traverses the route from the starting port to the end port, the project award paying for the cruise, etc. Specifically, there may be two different cruises conducted on the same vessel, but scheduled for different time periods and possibly traveling along different routes. The US academic research fleet currently possesses over 20 research vessels whose usage is shared and managed among 61 US academic institutions and national laboratories, all of which are members of the University-National Oceanographic Laboratory System (UNOLS).[8]

From a data integration perspective, the notion of cruise is also highly important as it acts as a “glue” that may connect all data about and results from the activities carried out during a cruise. This is also clearly reflected in the earlier example discovery scenario whereby, from GMRT data about a specified physiographic feature at some point-of-interest, one can obtain information about research programs relevant to the data. Hence, formalizing the notion of cruise would be an important step towards data integration as envisioned by the OceanLink project.

In this chapter, we describe an *ontology pattern* that formalizes the notion of oceanographic cruise, and further, how such a pattern can be used to help establishing cross-repository data integration, while respecting the heterogeneity existing in the different repositories involved in the OceanLink Project. The remainder of this chapter is organized as follows. Section 2 provides an overview of the OceanLink project and why we chose the ODP approach for data integration. We then present the formalization of an ODP for oceanographic cruises in Section 3 by first elaborating generic use cases guiding the design choices in specifying the pattern. Based on these generic use cases, we then formally specify the pattern in Section 4. This is then followed by a discussion in Section 5 on how the pattern can actually be used in applications, especially within the context of the OceanLink project. Finally, we close the chapter with a discussion on the relevant related work in Section 6.

[8] http://www.unols.org

2. The OceanLink Project and ODP

The ocean science community has decades of tradition of data sharing and openness advocated since the creation of Intergovernmental Oceanographic Commission (IOC) in 1960 [9]. With the advent of Linked Open Data, US ocean data repositories are in particular encouraged to adopt this new technology in publishing their data, greatly easing the data sharing. While this is far from finished, it is already apparent that cross-repository data discovery and integration is a nightmare since fundamental differences in data modeling exist among the repositories, hence simply publishing linked data is not sufficient [10].

The OceanLink project is an EarthCube effort to solve this problem by initiating a framework providing a horizontal integration amongst US ocean science data providers. This project, however, does not advocate the creation of a grand upper ontology for EarthCube, or ocean sciences because fundamental differences in data modeling and vocabularies between repositories, which occur due to differing subdomains, purposes and requirements, prevent such an ontology from being realized at all. Rather, the project opts for an approach using *ontology design patterns* (ODPs), which when coupled with good *community engagement*, is more likely to be successful in establishing the desired integrated framework.

An ODP is a reusable solution to some frequently occurring ontological modeling problem that emerges in different domains and can act as a building block for more complex ontologies [6]. The scope of modeling problems an ODP may address is quite broad, leading to different kinds of ODPs which are developed to solve them. This ranges from logical patterns which model certain logical constructs in a particular formal ontology language, to alignment patterns which act as templates representing commonly occurring types of alignments between ontologies, to content patterns which encapsulate generic notions within a particular domain of discourse. For data integration needs, content patterns are particularly useful for providing a unified perspective over the data while still permitting a rather significant degree of semantic independence between the data repositories. Concretely, each content pattern focuses only on one generic notion, realized as a self-contained, highly modular ontology that contains some axiomatization (preferably using a standard like OWL) that defines the formal semantics and relationships between the vocabulary items used in it. It represents what constitutes the given notion and what important and widely reusable aspects about it the domain experts have agreed upon. The axiomatization is carefully formulated such that no overly strong (i.e., application specific) ontological commitment is made by the pattern. In comparison to a monolithic, upper ontology, a content pattern can thus be seen as a snippet that defines only one particular notion without excessive intricacies an upper ontology may entail. Relationships to other patterns that define different, but related, notions can still be provided, but not specified in detail. Such characteristics make content patterns more suitable for heterogeneity preservation when integrating knowledge than monolithic foundational ontologies.

The ODP approach we employ for OceanLink leads to an architecture depicted in Figure 1, which is currently being implemented. This framework can be seen as a “hub” for ocean science data repositories. A collection of ODPs acts as a middle layer between the user interface and data repositories. The user interface translates user's requests into federated queries using vocabularies given by the patterns. Those queries are sent to appropriate data repositories, each of which is accompanied with a data-to-pattern alignment layer that translates the query in terms of its data model. The

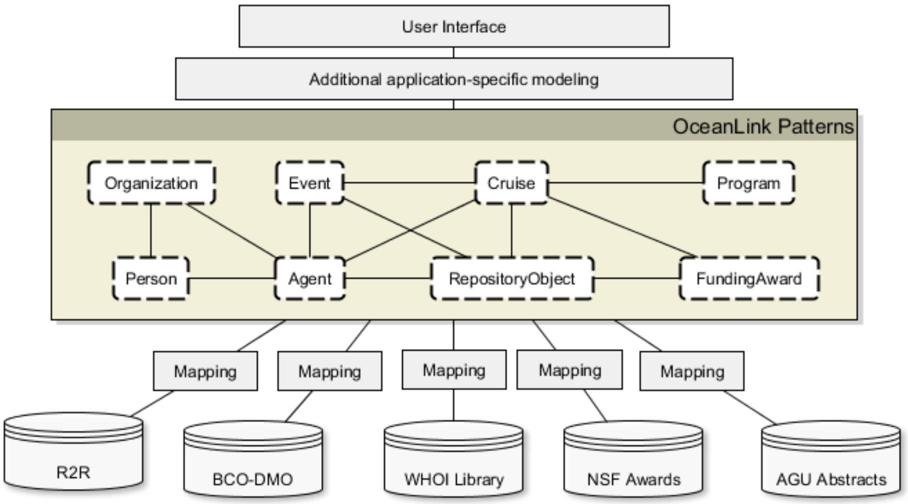


Figure 1: Architecture of OceanLink Cyberinfrastructure

OceanLink framework *does not force the adoption* of the patterns' vocabularies by the data repositories, but rather, ask each data repository to expose its content as an RDF dataset and *provide its own mapping* to the patterns as its alignment layer. Since the specification of the mapping is up to the data repositories, this scheme enables centralized discovery, while preserving heterogeneity of the data repositories. Currently, the following data are being integrated into the framework:

1. research vessels data from R2R;
2. biological and chemical ocean data from BCO-DMO;
3. cruise reports and PhD theses data from Marine Biological Laboratory Woods Hole Oceanographic Institution Library (MBLWHOI);
4. funded awards data from NSF; and
5. conference presentations and abstracts from American Geophysical Union (AGU).

With this flexible scheme, it is expected that more data repositories would join the framework in the future. Furthermore, the modularity of patterns in the ODP approach allows the vocabularies to be extended quite easily when necessary.

A crucial first step to realize this framework is the specification of the patterns. The modeling of the patterns is done through a series of interactive meetings with domain scientists and data repository maintainers, conducted in the style of the highly productive VoCamps[9]. As a result, we obtained more than a dozen patterns, which are currently being implemented. Amongst all those patterns, which include Person, Organization, Funding Award, etc., the Cruise pattern is rather special as it represents a notion that is rather specific for ocean sciences. Moreover, the modeling of the Cruise pattern presents us with an interesting case of reusing existing ontology patterns outside OceanLink, which provides us with a motivation for this paper.

[9] http://vocamp.org/wiki/Main_Page

3. Cruise: Generic Use Cases

Intuitively, the notion of oceanographic cruise is rather specific, since one can obviously also think of sight-seeing cruises, pleasure cruises, or even science cruises that are not used for ocean science purposes. From this perspective, to develop a pattern that is highly reusable even outside the ocean sciences, a generic notion of cruise would have to be modeled, rather than just the notion of oceanographic cruise. However, for the integration of oceanographic data within the OceanLink project, the more specific notion is adequate. Of course, rather than developing such a pattern from scratch, we will reuse, adjust, combine, and extend existing ontology patterns. This is done through established modeling practices while keeping the amount of abstract ontological commitments to a minimum.

For ocean science data repositories in OceanLink, a cruise can be seen as an abstract record that can act as a glue between otherwise separate pieces of information that ocean science data repositories may store. Those pieces of information are derived from generic use cases that guide which existing patterns we can reuse to develop the Cruise pattern. We describe such generic use cases through a number of *competency questions* that represent queries to the pattern.

One kind of competency question concerns the spatiotemporal information contained within the cruise route or trajectory. For example,

(Q1) "Find all cruises passing through Gulf of Maine in August 2013."

(Q2) "Show the trajectories of cruises in operation in September 2013."

Another kind of competency question involves querying the vessel on which a cruise is operated.

(Q3) "List all cruise vessels that departed from Woods Hole in 2012."

Also relevant to a cruise are competency questions for finding the people who serve in some capacity during the cruise's operation. For example,

(Q4) "Find the chief scientists of any cruise that collected samples of carbon-isotope data in Lake Superior."

Activities on a cruise may result in datasets or other digital objects stored in repositories, about which some users may issue questions such as:

(Q5) "What datasets were produced by the cruise AE0901?"

Finally, some party may also be interested in some administrative information about a cruise, exemplified by the following competency questions:

(Q6) "Which cruises are funded by the NSF award DBI-0424599?"

(Q7) "List all cruises under the Ocean Flux Program."

The above questions illustrate different pieces of information that are related to the notion of Cruise. From Question 1, 2, and 3, we know that trajectory and vessel are two important components of a cruise. A closer observation would lead us to an understanding that the trajectory and vessel of a cruise are indispensable: there is no cruise without a vessel and a trajectory. From Question 4, we understand that a cruise involves people who hold particular roles in its operation. To answer Question 5, information about an ocean science cruise clearly has to be related to the data and documents the cruise generated during its operation. Furthermore, due to Question 6 and 7, it also needs to be related to the information about the funding award and program which support the activities embodied by the cruise. In principle, all of these pieces of information are described by their own separate patterns which may possess more detailed information that need not be formulated explicitly in the cruise pattern.

4. Formalization in OWL

The use cases from Section 3 give us an insight that the notion of Cruise can essentially be viewed from three different angles: (1) as the route or trajectory a vessel traverses, hence providing the *spatiotemporal boundary* of a cruise; (2) as the collection of activities performed by *actors*, which can be humans or other kinds of agents; and (3) as a placeholder for various pieces of explanatory information that fit neither the trajectory nor the constituting activities, e.g., funding award, cruise type, etc. Points (1) and (2) motivate us to understand a cruise as a type of *event* since events are things that happen at some place and time whereby actors participate by performing some activities or roles. Moreover, by point (3), a cruise is not just a simple event; it is an event adorned with other explanatory information. Specifically, we conceptualize a cruise as *an adorned event undertaken by a vessel traversing through a particular trajectory*. This motivates a design choice where we formalize the Cruise pattern through reusing, adjusting, combining, and extending several already-existing patterns, including the Semantic Trajectory [11], Simple Event Model [12], and the Information Object pattern derived from DOLCE [13].

The following convention is used for all graphical depictions of the pattern in Figure 2, 3, 4, and 5. Rounded square nodes denote classes where a dotted line border means that the class also represents an external pattern whose details are unnecessary to specify within the Cruise pattern itself, i.e., they would be specified elsewhere in the definition of that pattern. Oval nodes denote instances defined explicitly in the pattern as controlled terms. All directed edges, except the ones labeled with `rdf:type` and `rdfs:subClassOf`, denote (object or data) properties where the direction is from the domain to the range of the denoted property. A dotted line means the property is defined in an external pattern.

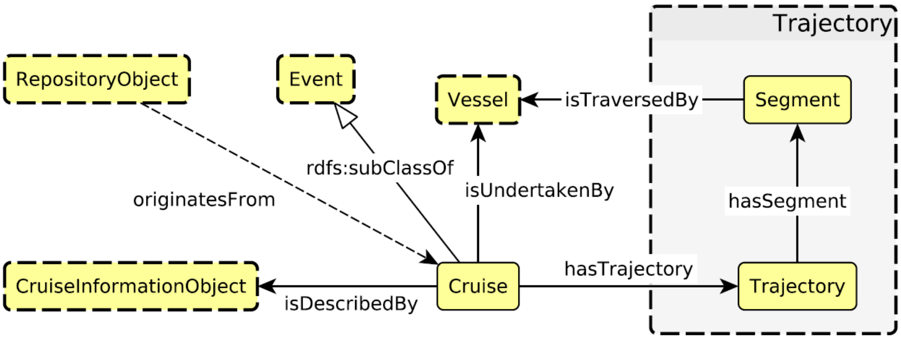


Figure 2: Overview of the Cruise pattern.

In addition to visual depictions (which remain somewhat ambiguous and cannot convey more complex relationships), the pattern is formalized as a set of axioms in the OWL 2 Web Ontology Language [14], which, for this chapter, are written in Manchester syntax [15]. In some places, we also employ a Datalog rule notation of the form B1 ∧ ... ∧ Bn → H where Bi's and H are atoms of the form C(*x*) or R(*x*, *y*) with C a class name and R a property name. Such a Datalog rule is understood as first-order implication whose variables are universally quantified, and there is a known translation from Datalog rules to DL axioms. Such a translation is out of scope of this paper and the reader may consult [16] for more details.

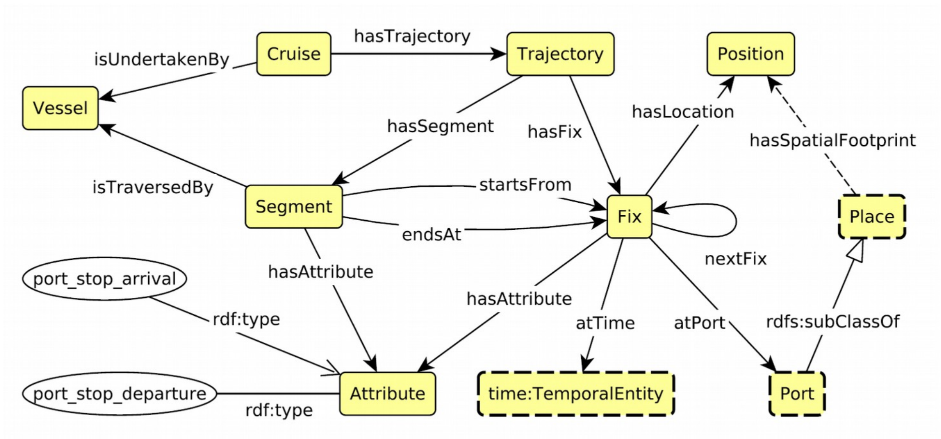
Figure 2 depicts a high level overview of the Cruise pattern, which omits some details explained and visualized in the remainder of this section. Notice that the relationship between the classes `Cruise`, `Trajectory`, and `Vessel` involves an internal class of the Trajectory subpattern.

Since a cruise is a kind of event, we specify that `Cruise` is a subclass of the more generic class `Event`. Adornments to the Cruise pattern are attached through an instance of the `CruiseInformationObject` class. In addition, Figure 2 also depicts a relationship between library digital objects (represented by the `RepositoryObject` class that covers datasets, papers, cruise logs, etc.) and cruises through the `originatesFrom` property. This property is not part of the Cruise pattern, but rather, defined in the RepositoryObject pattern, which is also being developed as part of the OceanLink project, though its specification is out of scope of this paper. Nonetheless, this relationship allows one to answer queries such as the one in Question 5.

4.1. Cruise Trajectory and Vessel

A trajectory is a sequence of spatiotemporal points of the form ⟨*x*, *y*, *t*⟩ (2D plane) or ⟨*x*, *y*, *z*, *t*⟩ (3D plane) where *x*, *y*, and *z* denote a coordinate on the plane and *t* denotes a time point. Those points are often generated by some moving object. A semantic trajectory is then usually understood as a trajectory in which the spatiotemporal points and segments (a pair of consecutive points) are adorned with useful geographic and domain knowledge allowing for more useful knowledge discovery.

There has been a large body of work in conceptualizing trajectory ([11,17,18,19], among others). Out of these many alternatives, the Semantic Trajectory pattern [11] is



```
(1)   Cruise SubClassOf: (hasTrajectory exactly 1 Trajectory)
(2)   Cruise SubClassOf: (isUndertakenBy exactly 1 Vessel)
(3)   Fix SubClassOf: (atTime some time:TemporalEntity)
(4)   Fix SubClassOf: (hasLocation some Position)
(5)   Fix SubClassOf: ((inverse hasFix) exactly 1 Trajectory)
(6)   Fix SubClassOf: (nextFix max 1 Fix)
(7)   Segment SubClassOf: (startsFrom exactly 1 Fix)
(8)   Segment SubClassOf: (endsAt exactly 1 Fix)
(9)   Segment SubClassOf: ((inverse hasSegment) some Trajectory)
(10)  (nextFix some owl:Thing) SubClassOf:
           ((inverse startsFrom) exactly 1 Segment)
(11)  ((inverse nextFix) some owl:Thing) SubClassOf:
           ((inverse endsAt) exactly 1 Segment)
(12)  endsAt SubPropertyChain: startsFrom o nextFix
(13)  Port SubClassOf: Place
(14)  port_stop_arrival Types: Attribute
(15)  port_stop_departure Types: Attribute
(16)  (hasAttribute value port_stop_arrival) SubClassOf: PortFix
(17)  (hasAttribute value port_stop_departure)
          SubClassOf: PortFix
(18)  hasLocation SubPropertyChain: atPort o hasSpatialFootprint
(19)  isUndertakenBy SubPropertyChain:
          hasTrajectory o hasSegment o isTraversedBy
```

Figure 3: Trajectory for Cruise

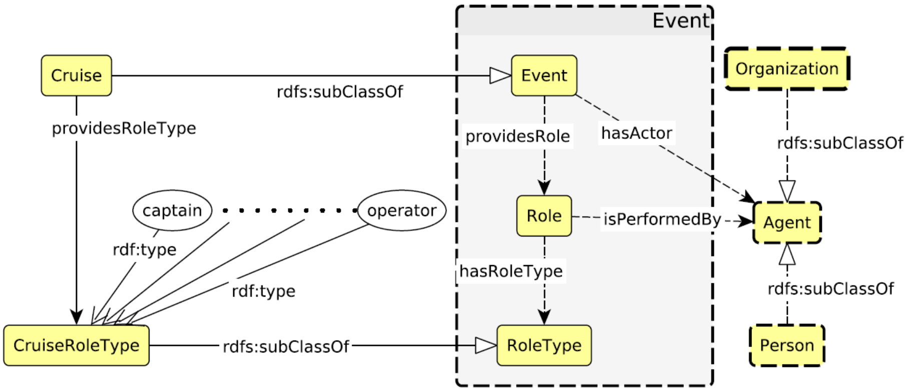
particularly chosen for reuse since it has a multi-granular conceptualization that makes it very versatile and reusable for many applications.

Reusing the Semantic Trajectory pattern as cruise trajectory leads us to the diagram and OWL axioms described in Figure 3. First of all, a trajectory and vessel are obviously two indispensable, interrelated parts of a cruise. In OceanLink, a cruise has exactly one trajectory and is undertaken by exactly one vessel, as formalized in axioms 1 and 2. Furthermore, the vessel must of course be the one that traverses the trajectory, which we formalize according to axioms 3–12, in addition to pairwise-disjointness between classes as well as domain and range restrictions for all properties here are asserted as discussed in the explanation of axioms 53–57 further below.

As in the Semantic Trajectory pattern, we define a cruise trajectory as a sequence of "points", called *fixes*, each of which possesses, at least, some position information and a timestamp. Generally, fixes and *segments* (pairs of consecutive fixes) can additionally be adorned with various geographic information and domain knowledge enabling a richer information discovery. Axioms 3–12 are similar to the ones in [11] – axioms 7, 8, and 9 in Figure 3 are in fact equivalent to axioms 2–5 of that paper. There is, however, an important difference leading to a slightly different axiomatization: the ordering of fixes in [11] using the `nextFix` property is entailed from the given two fixes and the corresponding segment; while here, the ordering is already explicit from the data and segments are auto-instantiated from it.

In our formalization, each fix has a position, a timestamp, and possibly some additional attributes; belongs to a trajectory; and is followed (through the `nextFix` property) by at most one other fix (axioms 3–6). Each segment starts from exactly one fix, ends at exactly one fix, and belongs to a trajectory (axioms 7–9). If a fix *x* is followed by another fix, then exactly one segment starts from *x* (axiom 9). Likewise, if a fix *x* is preceded by another fix, then exactly one segment ends at *x* (axiom 10). Axioms 9 and 10, however, do not guarantee that there is only one segment between two consecutive fixes. We can achieve this by ensuring that, whenever a segment *s* starts from a fix *x* whose next fix is *y*, then *s* must end at *y* (i.e., a rule of the form $\texttt{startsFrom}(x, y) \land \texttt{Fix}(y) \land \texttt{nextFix}(y, z) \rightarrow \texttt{endsAt}(x, z)$). Since there is exactly one segment ending at the fix *y* by axiom 11 and domain/range restrictions for the `startsFrom` and `nextFix` properties, the segment auto-instantiated by this axiom will be identified with *s*. The above rule can essentially be translated into a property chain axiom (the reader may consult [16] for further information how this can be done).

Position information attached to a fix can be, e.g., geospatial coordinates, and the position acts as an interface to richer geographic information about points-of-interest (POIs). For our need, we simply assume a generic class `Place` that represents a POI and has the position as its spatial footprint (realized through the `hasSpatialFootprint` property). Some of the fixes may be of particular interest as they represent ports where the cruise stops during its travel. A port is then here simply modeled as a kind of place (axiom 13). A fix corresponds to such a port if it has one of the following attributes: `port_stop_arrival` – when the fix's timestamp corresponds to the arrival time; and `port_stop_departure` – the fix's timestamp corresponds to the departure time (axioms 14–17). Also, the spatial footprint of the port gives us the fix's location (axiom 18). Finally, the vessel by which the cruise is undertaken must be the vessel that traverses the segments in the trajectory of the cruise (axiom 19). Note that in this modeling, vessel is only viewed as some class. In general, vessel should be modeled as its own pattern to accommodate richer information such as its size, type, and other useful information. In the context of cruise, however, those information are not so important, hence not included inside the cruise pattern.



```
(20)  (Role and ((inverse providesRole) some Event))
           SubClassOf: (hasRoleType exactly 1 RoleType)
(21)  (Role and ((inverse providesRole) some Event))
           SubClassOf: (isPerformedBy some Agent)
(22)  hasActor SubPropertyChain: providesRole o isPerformedBy
(23)  Cruise SubClassOf: Event
(24)  CruiseRoleType SubClassOf: RoleType
(25)  captain Types: CruiseRoleType
(26)  chief_engineer Types: CruiseRoleType
(27)  scientist Types: CruiseRoleType
(28)  cochief_scientist Types: CruiseRoleType
(29)  postdoc_scientist Types: CruiseRoleType
(30)  student Types: CruiseRoleType
(31)  graduate_student Types: CruiseRoleType
(32)  undergraduate_student  Types: CruiseRoleType
(33)  k12_student Types: CruiseRoleType
(34)  higher_ed_educator Types: CruiseRoleType
(35)  k12_educator Types: CruiseRoleType
(36)  technician Types: CruiseRoleType
(37)  marine_technician Types: CruiseRoleType
(38)  lead_marine_technician Types: CruiseRoleType
(39)  inspector Types: CruiseRoleType
(40)  observer Types: CruiseRoleType
(41)  foreign_observer Types: CruiseRoleType
(42)  other_observer Types: CruiseRoleType
(43)  scheduler Types: CruiseRoleType
(44)  operator Types: CruiseRoleType
(45)  other_role Types: CruiseRoleType
(46)  providesRoleType SubPropertyChain: rolifiedCruise
           o owl:topObjectProperty o rolifiedCruiseRoleType
(47)  Cruise EquivalentTo: rolifiedCruise Self
(48)  CruiseRoleType EquivalentTo: rolifiedCruiseRoleType Self
```

Figure 4: Cruise as a Kind of Event

4.2. Cruise as Event

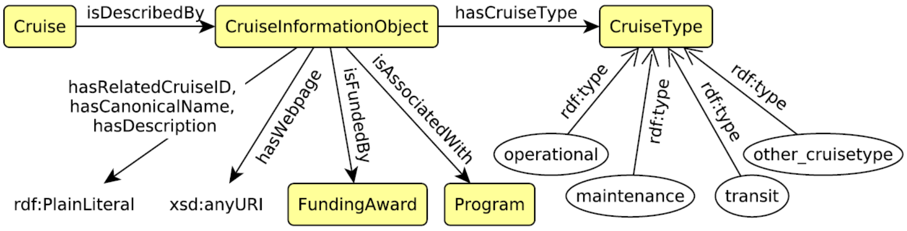
We realize the modeling of cruises as events (Figure 4) by reusing the Simple Event Model (SEM) [12]. This particular model is chosen because of its simplicity and the low ontological commitments within the model, especially in comparison to many other event models in the literature. As in SEM, an event consists of three essential components: place, time and actors. The grey rectangle within the figure represents an Event pattern inspired by SEM and covers the classes and properties that would have been defined there. Information about time and place is omitted there since for the Cruise pattern, they are already inherent within the trajectory. Any property within the Event pattern providing spatiotemporal information in this context can thus be written as a query on the trajectory information of the cruise.

We proceed with modeling the actors within a cruise. Note that SEM does not provide any OWL axiomatization, hence we also axiomatize the part of an event that concerns the actors. First, we do not enforce an event to always provide a role, but any role it provides must have exactly one type and be performed by some agent (axioms 20 and 21). Also, if a role provided by an event is performed by an agent, then this agent is an actor of the event (axiom 22). We make no assumption about agents except that people and organizations are considered agents, and these are asserted by the Person and Organization patterns whose description is out of scope of this paper.

Further, a cruise is an event (axiom 23) that also provides a predefined set of role types. For OceanLink, there are 20 cruise role types (captain, scientist, etc.), each of which is represented by a named individual (axioms 24–45). All cruise role types have to be provided by any cruise. This can be expressed as the rule `Cruise`(x) ∧ `CruiseRoleType`(y) → `providesRoleType`(x,y), and when translated into OWL, this becomes axioms 46, 47, and 48 where the properties `rolifiedCruise` and `rolifiedCruiseRoleType` are additional object properties needed to encode the atoms `Cruise`(x) and `CruiseRoleType`(y) in the above rule through the use of OWL's self-restriction and the predefined OWL object property `owl:topObjectProperty`, which is interpreted as a total binary relation connecting all pairs of individuals.

4.3. Cruise Information Object

Apart from spatiotemporal information and actor information, there are other explanatory pieces of information important for a cruise such as the funding award, cruise webpage, etc. These pieces of information are aggregated into an information object (Figure 5). Each cruise is then described by exactly one instance of such an information object (axiom 49). Most explanatory information is optional, except that exactly one cruise type is required for each cruise information object (axiom 50) and the set of cruise types is predefined (axiom 51). Finally, in the OceanLink context, a cruise is operational if, and only if, it has a chief scientist and is funded by some funding award. This expressed by axiom 52.



```
(49)  Cruise SubClassOf:
         (isDescribedBy exactly 1 CruiseInformationObject)
(50)  CruiseInformationObject SubClassOf:
         (hasCruiseType exactly 1 CruiseType)
(51)  CruiseType EquivalentTo:
         {  operational, transit, maintenance, other_cruisetype }
(52)  Cruise and
            isDescribedBy some (hasCruiseType value operational)
         EquivalentTo:
            providesRole some
               (Role and (hasRoleType value chief_scientist))
            and (isFundedBy some FundingAward)
```

Figure 5: Cruise Information Object

4.4. Class Pairwise-Disjointness, and Domain and Range of Properties

We assert that all classes in the pattern are pairwise disjoint, except for each of the following pairs: (`Cruise`, `Event`), (`Port`, `Place`), and (`CruiseRoleType`, `RoleType`) – each of which is a subclass-superclass pair. The following axiom exemplifies pairwise-disjointness of `Cruise` and `Vessel`.

```
(53) DisjointClasses: Cruise, Vessel
```

Also, we enforce the unique name assumption is for all named individuals, e.g., `port_stop_arrival` and `port_stop_departure` refer to different individuals. In addition, we assert *guarded* domain and range restrictions for *all* of the object and data properties in the pattern as exemplified for the `hasFix` object property and `hasRelatedCruiseID` data property below:

```
(54)  (hasFix some Fix) SubClassOf: Trajectory
(55)  Trajectory SubClassOf: (hasFix only Fix)
(56)  hasRelatedCruiseID some rdf:PlainLIteral
            SubClassOf: CruiseInformationObject
(57)  CruiseInformationObject SubClassOf:
            hasRelatedCruiseID only rdf:PlainLiteral
```

The domain and range restrictions in the form as above constitute weaker ontological commitments than the usual domain and range restriction, i.e., the ones of the form `P rdfs:domain C` and `P rdfs:range D` where `C` and `D` are resp. the domain and range of P. These last two are equivalent to the axioms (P some owl:Thing) SubClassOf: C and owl:Thing SubClassOf: P only D, since they constitute very strong ontological commitments that are not required for the modeling; thus we stick to good modeling practice and guard domains and ranges.

4.5. Views for the Cruise Pattern

In summary, the Cruise pattern glues together three existing patterns: Trajectory, Event, and Information Object. This combination may make the Cruise pattern a bit complicated, both for data providers as well as for users. To aid them in readability and ease of use, it is often useful to specify some semantic "shortcuts" that capture some common queries over the pattern. Such shortcuts, called \emph{views}, can be defined depending on application needs and typically expressed as rules that can be translated into OWL axioms (e.g., how axioms 46, 47, and 48 were obtained). For example, the hasChiefScientist property connects a cruise and its chief scientist:

(58) $\texttt{Cruise}(x) \wedge \texttt{providesRole}(x, y) \wedge \texttt{isPerformedBy}(y, z)$
$\wedge\ \texttt{Person}(z) \wedge \texttt{hasRoleType}(y, \texttt{chief\_scientist})$
$\rightarrow \texttt{hasChiefScientist}(x, z)$

Another example is the starting port of a cruise (the ending port is similar), obtained from axiom 59 and 60.

```
(59)  Fix and ( not (inverse endsAt) some Segment )
          SubClassOf: StartingFix
```

(60) $\texttt{Cruise}(x) \wedge \texttt{hasTrajectory}(x, y) \wedge \texttt{hasFix}(y, z) \wedge \texttt{startingFix}(z)$
$\wedge\ \texttt{atPort}(z, p) \rightarrow \texttt{hasStartingPoint}(x, p)$

Such views can easily be defined depending on the application needs. More importantly, they can also be made available either within the same OWL ontology containing the pattern, or in a separate OWL ontology that imports the pattern.

5. Application Scenarios

The Cruise pattern, together with the other OceanLink patterns, will be deployed as a middle layer – the *pattern layer* – according to Figure 1, and hosted in the OceanLink's own server. The OWL serialization of the pattern can be found at http://schema.oceanlink.org. Each data source then types its instance data against the classes and properties in the patterns. For example, in the R2R repository, all cruises are typed (via rdf:type) as r2r:Cruise, while in the BCO-DMO repository, all cruises are instances of bcodmo:Deployment with platform type bcodmo:Vessel. The data provider's task is then to ensure that their cruise instances would also be typed as Cruise from the pattern. The following two SPARQL queries for R2R and BCO-DMO achieve this where the CONSTRUCT clause generates a set of triples of the form ?x rdf:type :Cruise.

```
CONSTRUCT ?x rdf:type :Cruise
WHERE { ?x rdf:type r2r:Cruise. }

CONSTRUCT ?x rdf:type :Cruise
WHERE { ?x a bcodmo:Deployment;
        bcodmo:ofPlatform [a bcodmo:Vessel]. }
```

Using these kinds of SPARQL queries applied to both classes (e.g., `:Cruise` and properties in the patterns, each data repository produces a *derived graph* (from the set of triples formed by the `CONSTRUCT` clause) that can be aggregated and cached at the pattern layer. Such a derived graph intuitively projects the data from a repository according to the structure specified by the patterns, hence realizing the mapping from the data to the pattern. With this in place, a user can then issue a query via the user interface using only vocabularies defined in the patterns.

Suppose one is interested in finding all ports at which the researcher named “Mak Saito” stopped by in any of his expeditions. This can be expressed as the following SPARQL query over the Cruise pattern as follows, assuming `:hasLegalName` is a property defined in the Person pattern:

```
SELECT ?port WHERE {
  ?port a :Port.
  ?cruise :hasTrajectory ?t ;
          :hasActor ?x.
  ?t :hasFix ?f.
  ?f :atPort ?port.
  ?x rdf:type :Person; :hasLegalName "Mak Saito". }
```

For another example, suppose one wishes to find out who joined any cruise that went through the Gulf of Maine, what their role was in the cruise, and what funding award supported their trip. This can be expressed using SPARQL as:

```
SELECT ?name ?role ?fund WHERE {
  ?cruise :isDescribedBy ?d; :providesRole ?r;
          :hasFix ?x.
  ?d :isFundedBy ?f.
  ?f :hasAwardID ?fund.
  ?r :hasRoleType ?role; :isPerformedBy ?p.
  ?p rdf:type :Person; :hasLegalName ?name.
  ?x :hasLocation ?pos.
  ?pl :hasSpatialFootprint ?pos; rdfs:label ?pln.
  FILTER regex(?pln, "Gulf of Maine", "i").
```

Clearly, satisfactory answers to such queries depend on the completeness of the derived graphs are constructed by the data provider. A rather crude alignment using SPARQL's `CONSTRUCT` clause above works only when a straightforward correspondence between the data and the patterns can be obtained. Otherwise, more expressive alignment schemes, possibly involving complicated inferencing may need to be employed, and further discussion on how such inferencing can be done is out of scope for this paper. On the other hand, this alignment-based approach is highly flexible because if a new data source needs to be added, the data provider simply has to establish the alignment to the pattern. Furthermore, there is no obligation for the data providers to completely specify such an alignment to every vocabulary in the patterns. It is up to them to choose which vocabulary items in the patterns they want to map to. The only consequence is that the less complete their alignment is, the less data can be discovered from their repositories.

6. Related Work

As far as we know, we are not aware of existing works specifically addressing the conceptual modeling of oceanographic cruises. However, there has been a considerable body of work concerning conceptualization of semantic trajectories and events. We briefly discuss some of the existing work that are most closely related here.

Recall that our cruise trajectory model follows [11] quite closely. The only differences are on the slightly more involved relationship between the moving object (i.e., vessel), the trajectory and segments of the cruise trajectory, as well as on the way some of the axioms were defined in order to satisfy OceanLink specific requirements and settings.

The first model of semantic trajectory itself were introduced in [18]. Here, a semantic trajectory consists of two facets: geometric and semantic. The geometric facet is realized through the notions of stop and move, which are analogous to fix and segment. Semantic facet is realized through annotating stops and moves. The follow-up work in [19] built on [18] by proposing the Trajectory ontology, which is essentially a big combination of geometric trajectory ontology, geography ontology and application domain ontology. This conceptualization, however, seems quite involved requiring multistep process in the construction. The construction itself is rather ad hoc, especially the geography ontology and application domain ontology, both of which are application dependent. The resulting Trajectory ontology may then suffer from too much rigidity and overly strong ontological commitments. Our semantic trajectory (and that of [11]) is, in this aspect, more flexible and thus more suitable for data integration scenario within OceanLink.

Bogorny, et al. [17] proposed a data model for semantic trajectory, putting a stronger emphasis on the raw trajectory data. They generalized [18] by introducing the notion of sub-trajectory, which can be seen as a construct on the data level, as well as adding a number of aspects, including Transportation Means, Goal, and Behavior. Transportation Means describes the means of transportation taken by the object to move along a certain part of the trajectory. Goal represents the reason why the object moves. Behavior aspect describes a set of characteristics that distinguish the trajectory of a moving object, which is typically computed through some intelligent methods or mining algorithms. The whole aspects, however, do not appear to have a clear formalization, which is probably expected given the emphasis on the data level. On the other hand, our cruise trajectory was designed from the start to be generic, and can cover the above aspects. Transportation Means are modeled through moving objects, i.e., Vessel in our case. Goals and Behaviors can be covered by adding or specializing attributes of fixes and segments, or attaching new classes to the Trajectory class.

Meanwhile, regarding events, the existing body of work is even more extensive. For example, the Event Ontology [20], Linking Open Description of Events (LODE)

[21], the F-model [22], the ABC Ontology [23], and SEM [12]. These models differ in terms of scope, domain-dependency, focus, as well as formalization, due to the different purposes they were intended to. For our purpose of the Cruise pattern, SEM was eventually chosen as the basis of our event modeling due to its simplicity and flexibility. Even then, the reuse from SEM is is actually more of "getting inspiration" from SEM than importing the actual signature or conceptualization. The key point in the reuse is the idea from SEM that an event simply consists of actors and spatiotemporal information. This idea fits perfectly with the notion of patterns that we use for data integration while preserving heterogeneity as much as possible.

7. Conclusion

In this paper, we presented an ontology design pattern for oceanographic cruises. We showed how this pattern was specified as a combination and reuse of existing patterns: trajectory, event, and information object. We then demonstrated the applicability of this pattern in integrated knowledge scenarios within the OceanLink project and also argued for the general reusability of the pattern.

One direction for future work is regarding the actual implementation and application of this pattern within OceanLink's integrated knowledge discovery service that is currently being implemented. In particular, we plan to study the effectiveness and ease of use from a user's perspective in serving a variety of information needs through the OceanLink service. From the data providers' perspective, we will study the ease of use in aligning their data to the pattern as well as the practical extensibility and reusability of the pattern.

Another direction for future work beyond the OceanLink project that we also wish to pursue concerns a number of more fundamental, theoretical questions arising from our experience in specifying and implementing this pattern. This includes, among others, problems regarding the use of such a pattern for data integrity; the expressivity and computational issues surrounding views; and, how flexible the pattern is for data integration, especially if one wishes to add new data sources.

Acknowledgement. The work presented is this chapter has been primarily funded by the National Science Foundation under the OceanLink project, award 1354778 *EAGER: Collaborative Research: EarthCube Building Blocks, Leveraging Semantics and Linked Data for Geoscience Data Sharing and Discovery.* Any opinions, findings, and conclusions expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

[1] Tanu Malik and Ian Foster. 2012. "Addressing Data Access Needs of the Long-Tail Distribution of Geoscientists." In *Geoscience and Remote Sensing Symposium (IGARSS), 2012 IEEE International*, pages 5348–5351.

[2] Amanda L. Mascarelli. 2009. "Data's Shameful Neglect." *Nature*, 461(7261):145.

[3] Committee on an Ocean Infrastructure Strategy for U.S. Ocean Research in 2030; National Research Council. 2011. *Critical Infrastructure for Ocean Research and Societal Needs in 2030*. The National Academies Press.

[4] William B.F. Ryan, Suzanne M. Carbotte, Justin O. Coplan, Suzanne O'Hara, Andrew Melkonian, Robert Arko, Rose Anne Weissel, Vicki Ferrini, Andrew Goodwillie, Frank Nitsche, Juliet Bonczkowski, and Richard Zemsky. 2009. "Global Multi-Resolution Topography Synthesis." *Geochemistry, Geophysics, Geosystems*, 10(3).

[5] Christian Bizer, Tom Heath, and Tim Berners-Lee. 2009. "Linked Data – The Story So Far." *International Journal on Semantic Web and Information Systems*, 5(3):1–22.

[6] Aldo Gangemi. 2005. "Ontology Design Patterns for Semantic Web Content." In Yolanda Gil, Enrico Motta, V. Richard Benjamins, and Mark A. Musen, editors, *The Semantic Web – ISWC 2005, 4th International Semantic Web Conference, ISWC 2005, Galway, Ireland, November 6-10, 2005, Proceedings*, volume 3729 of Lecture Notes in Computer Science, pages 262–276. Springer.

[7] Krzysztof Janowicz and Pascal Hitzler. 2012. "The Digital Earth as Knowledge Engine." *Semantic Web*, 3(3):213–221.

[8] Committee on Evolution of the National Oceanographic Research Fleet; National Research Council. 2009. *Science at Sea: Meeting Future Oceanographic Goals with a Robust Academic Research Fleet*. The National Academies Press.

[9] David M. Glover, Peter H. Wiebe, Cynthia L. Chandler, and Sydney Levitus. 2010. "IOC Contributions to International, Interdisciplinary Open Data Sharing." *Oceanography*, 23(3):140–151.

[10] Sean Bechhofer, Iain E. Buchan, David De Roure, Paolo Missier, John D. Ainsworth, Jiten Bhagat, Philip A. Couch, Don Cruickshank, Mark Delderfield, Ian Dunlop, Matthew Gamble, Danius T. Michaelides, Stuart Owen, David R. Newman, Shoaib Sufi, and Carole A. Goble. 2013. "Why Linked Data is Not Enough for Scientists." *Future Generation Computer Systems*, 29(2):599–611.

[11] Yingjie Hu, Krzysztof Janowicz, David Carral, Simon Scheider, Werner Kuhn, Gary Berg-Cross, Pascal Hitzler, Mike Dean, and Dave Kolas. 2013. "A Geo-Ontology Design Pattern for Semantic Trajectories." In Thora Tenbrink, John G. Stell, Antony Galton, and Zena Wood, editors, *Spatial Information Theory – 11th International Conference, COSIT 2013, Scarborough, UK, September 2-6, 2013, Proceedings*, volume 8116 of Lecture Notes in Computer Science, pages 438–456, Springer.

[12] Willem Robert van Hage, Véronique Malaisé, Roxane Segers, Laura Hollink, and Guus Schreiber. 2011. "Design and Use of the Simple Event Model (SEM)." *Journal of Web Semantics*, 9(2):128–136.

[13] Daniel Oberle, Anupriya Ankolekar, Pascal Hitzler, Philipp Cimiano, Michael Sintek, Malte Kiesel, Babak Mougouie, Stephan Baumann, Shankar Vembu, Massimo Romanelli, Paul Buitelaar, Ralf Engel, Daniel Sonntag, Norbert Reithinger, Berenike Loos, Hans-Peter Zorn, Vanessa Micelli, Robert Porzel, Christian Schmidt, Moritz Weiten, Felix Burkhardt, and Jianshen Zhou. 2007. "DOLCE ergo SUMO: On Foundational and Domain Models in the SmartWeb Integrated Ontology (SWIntO)." *Journal of Web Semantics*, 5(3):156–174.

[14] Pascal Hitzler, Markus Krötzsch, Bijan Parsia, Peter F. Patel-Schneider, and Sebastian Rudolph, editors. 2009. "OWL 2 Web Ontology Language: Primer.

W3C Recommendation 27 October 2009." Available from http://www.w3.org/TR/owl2-primer/.

[15] Matthew Horridge and Peter F. Patel-Schneider, editors. 2009. "OWL 2 Web Ontology Language: Manchester Syntax. W3C Recommendation, 27 October 2009." Available from http://www.w3.org/TR/owl2-manchester-syntax/.

[16] Adila Krisnadhi, Frederick Maier, and Pascal Hitzler. 2011. "OWL and Rules." In Axel Polleres, Claudia d'Amato, Marcelo Arenas, Siegfried Handschuh, Paula Kroner, Sascha Ossowski, Peter F. Patel-Schneider, editors, *Reasoning Web. Semantic Technologies for the Web of Data – 7th International Summer School 2011, Tutorial Lectures*, volume 6848 of Lecture Notes in Computer Science, pages 382–415. Springer.

[17] Vania Bogorny, Chiara Renso, Artur Ribeiro de Aquino, Fernando de Lucca Siqueira, and Luis Otávio Alvares. 2014. "CONSTAnT – A Conceptual Data Model for Semantic Trajectories of Moving Objects." *Transactions in GIS*, 18(1):66–88.

[18] Stefano Spaccapietra, Christine Parent, Maria Luisa Damiani, José Antônio Fernandes de Macédo, Fabio Porto, and Christelle Vangenot. 2008. "A Conceptual View on Trajectories." *Data & Knowledge Engineering*, 65(1):126–146.

[19] Zhixian Yan, Jose Macedo, Christine Parent, and Stefano Spaccapietra. 2008. "Trajectory Ontologies and Queries." 2008. *Transactions in GIS*, 12(s1):75–91.

[20] Yves Raimond and Samer Abdallah. 2007. "The Event Ontology." Available from http://purl.org/NET/c4dm/event.owl.

[21] Ryan Shaw, Raphaël Troncy, and Lynda Hardman. 2009. "LODE: Linking Open Descriptions of Events." In Asunción Gómez-Pérez, Yong Yu, Ying Ding, editors, *The Semantic Web, Fourth Asian Conference, ASWC 2009, Shanghai, China, December 6-9, 2009, Proceedings*, pages 153–167, Springer.

[22] Ansgar Scherp, Thomas Franz, Carsten Saatho, and Steffen Staab. 2009. "F – A Model of Events Based on the Foundational Ontology DOLCE+DnS Ultralight." In Yolanda Gil, Natasha Fridman Noy, editors, *Proceedings of the 5th International Conference on Knowledge Capture (K-CAP 2009), September 1-4, 2009, Redondo Beach, California, USA*, pages 137–144. ACM.

[23] Carl Lagoze and Jane Hunter. 2001. "The ABC Ontology and Model." *Journal of Digital Information*, 2(2).